

final draft

Lecture 10 - Achievements and Gameplay

"I don't care if it works on your machine! We are not shipping your machine!" - Vidiu Platon

- ▶ how can we integrate an achievement system into an engine?
- ▶ how can we implement in-game tutorials, side quests, quick-time events?

Last Week Recap

- ▶ Scripting and domain-specific languages

Why Achievements?

- ▶ Gamification: research shows that gamification can keep players engaged with repetitive tasks

This can include achievements, badges, trophies, levelling up, more powerful items, etc. Borderlands series is a good example. Various Maths tricks used in RPGs can also help.

Quests

Quest systems typically utilise a form of gamification where the player is generously rewarded upon completing a quest.

Quest Design Demo

Let's design a quest for Skyrim.

Consider: NPCs, rewards, story. Group into teams?

Activity

Quest Management System using the quest we designed.

1. How can we design a data structure for storing quests as external data? What do we need to store?
2. Think about serialization. Can we specify quests using external data formats, e.g. json.

Quick-time Events

Commonly used to create an exciting action sequence with dynamic camera movement, possibly altering main gameplay.

Quick-time Events (Impl)

1. Check for input
2. Design a sequence of input events and check input matches

`sequence = { Key.W, Key.D, Key.W, Key.S }`

```
for (key : sequence) {  
    if (key != input_key) {  
        break;  
    }  
}
```

In-game Tutorials

- ▶ Help the player learn the game
- ▶ Needs to be designed carefully to avoid bombarding with new knowledge

Activity

1. In-game Tutorials: design a simple tutorial for an RPG, such as Skyrim. What will you be teaching the player?
2. Think in terms of data structures you designed. How are you going to store the tutorial as external data.

All About Numbers

- ▶ Players like progress in any form but especially numerical
- ▶ Hero level, hit points, currency, number of items, etc.
- ▶ Mechanic: dice roll, combat maths, economics
- ▶ Any other examples?

All About Numbers

Example: side quests.

Players will still take those even if not relevant to the story. Primary reason: good (rare) rewards.

All About Numbers

Obtaining rare items by chance. Example: refine an item with 1 metal to gain +10 damage. Success chance: 60%. If the item is not refined, it will be broken.

Activity

Design and implement a simple “refine item” logic. At level 1, the chance to upgrade to level 2 is 100%. Then 2->3 is 90%, 3->4 is 80% and so on.

If the item breaks, create a new one.

Gameplay type

What do you want the player to do?

- ▶ Intellect-driven (strategise and analyse)
- ▶ Skill-driven (refine common repetitive actions)
- ▶ Casual (time killer)

Engagement

Define clear win and lose conditions. Players will engage quicker.
However, pay attention to difficulty: too easy / too difficult.

Balance

Allow new players to have a chance of defeating old (experienced) players. Examples: reduced stats for experienced players, leaderboard reset, PvE adjustment based on player level. Alternative solutions: matchmaking.

Player Experience

- ▶ Sound effects / music - easy way to add immersion
- ▶ Story: game world state is bad, player appears, player actions make the state better, villain reveals his secret power, state is bad again, player learns new power, player wins, state is now good

Activity

1. Find a game that you like, consider why you like it.
2. Can you implement a particular gameplay feature in your game?

Conclusion

- ▶ Gameplay is at the same time the easiest and the hardest aspect of the game being developed.
- ▶ Standard practices exist.

Tutorial

Assignment work